Lookahead Pathology in Real-Time Path-Finding

Vadim Bulitko
University of Alberta, Department of Computing Science
Edmonton, Alberta, Canada T6G 2E8
bulitko@ualberta.ca
phone: +1 (780) 492-3854
http://dis.ijs.si/Mitjal/supplements/aaai06-mp.htm

Mitja Luštrek
Jožef Stefan Institute
Department of Intelligent Systems
Jamova 39, 1000 Ljubljana, Slovenia
mitja.lustrek@ijs.si

Type: original, unpublished work

Introduction

Path-finding tasks commonly require real-time response, which on large problems precludes the use of complete search methods such as A*. Incomplete single-agent search methods work similarly to minimax-based algorithms used in two-player games. They conduct a limited-depth lookahead search, i.e., expand a part of the space centered on the agent, and heuristically evaluate the distances from the frontier of the expanded space to the goal.

Actions selected based on heuristic lookahead search are not necessarily optimal, but both in minimax search and in single-agent search it is generally believed that deeper lookahead increases the quality of decisions. Theoretical analyses of minimax have shown that the opposite is sometimes the case (Nau 1979; Beal 1980). This phenomenon has been termed the minimax pathology. Recently pathological behavior was observed in single-agent search as well (Bulitko *et al.* 2003; Bulitko 2003).

In this poster we investigate lookahead pathology in real-time path-finding on maps from commercial computer games. Pathology was experimentally observed in *more than half* of the 1,000 problems considered. This indicates that in single-agent search, pathological behavior is a practical issue - unlike in minimax search, where it seems to be mostly an artifact of the theoretical analyses.

The Pathology

The setting of this poster is a two-dimensional grid world, in which an agent is trying to find a path from a start to a goal state. The agent plans its path using the Learning Real-Time Search (LRTS) algorithm (Bulitko & Lee 2006) configured so that it works similarly to the classic LRTA* (Korf 1990). LRTS conducts a lookahead search centered on the current state and generates all the states within its lookahead area (i.e., up to d moves away from the current state). It then heuristically estimates the distances from the frontier states to the goal state and moves to the most promising frontier state. Upon reaching it, it conducts a new search. The initial heuristic is the actual shortest distance assuming an empty map. After each search, the heuristic of the current state is updated to the estimated distance through the most promising frontier state, which constitutes the process of learning.

The most natural way to measure the performance of a search algorithm is the length of the path it finds. This can be used in *on-policy* experiments, where the agent follows a path from the start state to the goal state as directed by the LRTS algorithm, updating the heuristic along the way. However, we also conducted *off-policy* experiments, where the agent is teleported into a (randomly selected) state, in which it selects the first move towards the goal state; the heuristic is not updated. Such experiments require a different measure of performance.

In any state s from a set of states S, the agent can take an optimal or a suboptimal action. An optimal action moves the agent into a state lying on the shortest path from s to the goal state. The *error* $e(S, d)$ is the fraction of suboptimal actions taken in the set of states S using lookahead depth d. The degree of *error pathology* in S is k iff $e(S, i+1) > t \cdot e(S, i)$ for k different i. The tolerance t was introduced to reduce effects of random noise and is discussed at the URL above.

First we conducted the basic on-policy experiment. The on-policy results in Table 1 show a definite presence of pathology. The **first explanation** is that our maps contain a lot of intrinsically pathological states. This was verified by the basic off-policy experiment, where the error was measured on randomly selected states. The off-policy results in Table 1 indicate that the first explanation is *not* correct.

Table 1: Percentage of pathological problems in the basic on-policy and off-policy experiments.

Degree	No pathology	1	2	3	4	≥ 5
On-policy	42.3	19.7	21.2	12.9	3.6	0.3
Off-policy	95.7	3.7	0.6	0.0	0.0	0.0

Why the large difference between the results of the basic on-policy and off-policy experiments? The basic on-policy experiment involves learning and the basic off-policy experiment does not. On-policy, the agent performs a search with lookahead depth d every d moves. If the map were empty, the only updated state it would encounter during each search would be the one updated during the previous search. Each search generates $(2d+1)^2$ distinct states, so $1/(2d+1)^2$ of them would have been updated: a fraction that is larger for smaller d. We can now formulate the **second explanation**. Smaller lookahead depths benefit more from the updates to the heuristic. This can be expected to make their decisions better than the mere depth would suggest and thus closer to larger lookahead depths. If they are closer to larger lookahead depths, cases where a deeper lookahead actually per-

forms worse than a shallower one should be more common.

The second explanation can be verified in two ways: (1) by conducting the basic on-policy experiment and measuring the error separately without taking into account the updates to the heuristic and (2) by measuring the volume of updates, i.e., the sum of the differences between the updated and the initial heuristic. For the first way, Table 2 shows much less pathology than in the basic on-policy experiment, suggesting that learning is indeed responsible for the pathology. For the second way, the volume of updates in the basic on-policy experiment decreases from 4.1 at $d = 1$ to 1.4 at $d = 10$, so it is larger at smaller lookahead depths as predicted. There are no updates in the basic off-policy experiment.

Table 2: Percentages of pathological problems in an on-policy experiment with error measured without learning.

Degree	No pathology	1	2	3	4	≥ 5
Problems	79.8	14.2	4.5	1.2	0.3	0.0

The results in Table 2 still show more pathology than in the basic off-policy experiment, so there must be another reason for pathological behavior. This leads us to the **third explanation**. Let $\alpha_{\text{off}}(d)$ and $\alpha_{\text{on}}(d)$ be the average number of states generated per move in the basic off-policy and on-policy experiments correspondingly. In on-policy experiments a search is performed every d moves, so $\alpha_{\text{on}}(d) = \alpha_{\text{off}}(d)/d$. This means that lookahead depths in the basic on-policy experiment are closer to each other with respect to the number of states generated than in the basic off-policy experiment. Since the number of states generated can be expected to correspond to the quality of decisions, cases where a deeper lookahead actually performs worse than a shallower lookahead should be more common.

The third explanation can also be verified in two ways: (1) by an on-policy experiment where a search is performed every move instead of every d moves and (2) by measuring the number of states generated. For the first way, Table 3 supports the third explanation: the percentage of pathological problems is considerably smaller than in the basic on-policy experiment. For the second way, the number of states generated is around 8 at $d = 1$ in all types of experiments. In the basic off-policy experiment and in the on-policy experiment when searching every move, it increases to around 1,550 at $d = 10$. In the basic on-policy experiment, it increases to the much smaller 146.3 at $d = 10$ as predicted.

Table 3: Percentages of pathological problems in an on-policy experiment when searching every move.

Degree	No pathology	1	2	3	4	≥ 5
Problems	86.9	9.0	3.3	0.6	0.2	0.0

Towards a Remedy

We have shown that pathological behavior is common in real-time path-finding. We must now consider how much an LRTS agent would benefit were it able to select optimal lookahead depth dynamically. The best fixed lookahead depth for our 1,000 problems is 1 and results in the average path length of 175.4. If the optimal lookahead depth is selected for each problem (i.e., for each start state), the average path length decreases to 107.9. The improvement in path length is quite significant and motivates research into automated methods for lookahead depth selection.

The most straightforward way to select the optimal lookahead depth is to pre-compute the path lengths for all lookahead depths for all states on the map. We did this for an example map of an office space and a garden, which can be viewed at the URL listed in the front matter of the paper. The best fixed lookahead depth for this map is again 1 and results in the average path length of 253.2. If the optimal lookahead depth is selected for each problem, the average path length decreases to 132.4. This is similar to the 1,000 problems, indicating that the map is reasonably representative. Once the optimal lookahead depth is known for each state, we can go even further: we can adapt the lookahead depth *per move*. This approach gives the average path length of 113.3 and also decreases the average number of nodes generated per move from 59.3 to 34.0.

Determining the optimal lookahead depth for every pair of states on the map is computationally very expensive: the 8,743-state office-and-garden map contains over 7.6×10^7 directed pairs of states. We can make the task more tractable with state abstraction such as the clique abstraction (Bulitko, Sturtevant, & Kazakevich 2005). In this approach cliques of states are merged into single abstract states. For each abstract state s_a, we find a ground-level state s_{gl} closest to the average coordinates of states abstracting into s_a. We then compute the optimal lookahead depth for s_{gl} and associate it with s_a. During the on-line search, the lookahead depth for the agent's state is retrieved from the state's abstract parent. If appropriate abstraction is used in the office-and-garden map, we can obtain the average path length of 169.3 and need to pre-compute the optimal depths for only 0.004% of all state pairs. Similar performance is expected on other comparable maps, but further investigation is required.

References

Beal, D. F. 1980. An analysis of minimax. In Clarke, M. R. B., ed., *Advances in Computer Chess*, volume 2, 103–109. Edinburgh, UK: Edinburgh University Press.

Bulitko, V., and Lee, G. 2006. Learning in real time search: A unifying framework. *JAIR* 25:119–157.

Bulitko, V.; Li, L.; Greiner, R.; and Levner, I. 2003. Lookahead pathologies for single agent search. In *Proceedings of IJCAI, poster section*, 1531–1533.

Bulitko, V.; Sturtevant, N.; and Kazakevich, M. 2005. Speeding up learning in real-time search via automatic state abstraction. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, 1349–1354.

Bulitko, V. 2003. Lookahead pathologies and meta-level control in real-time heuristic search. In *Proceedings of the 15th Euromicro Conference on Real-Time Systems*, 13–16.

Korf, R. E. 1990. Real-time heuristic search. *Artificial Intelligence* 42(2, 3):189–211.

Nau, D. S. 1979. *Quality of Decision versus Depth of Search on Game Trees*. Ph.D. Dissertation, Duke U.